

**REMARKS**

Applicant respectfully requests further examination and reconsideration in view of the comments set forth fully below. Claims 1-25 were previously pending. Within the Office Action, Claims 1-25 have been rejected. Accordingly, Claims 1-25 are currently pending.

**Rejections Under 35 U.S.C. § 103**

Within the Office Action, Claims 1-13, 15-18, and 20-24 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2003/0182450 to Ong et al. (hereafter “Ong”) in view of SyncML Sync Protocol, version 1.0.1, [http://www.openmobilealliance.org/tech/affiliates/syncml/syncml\\_protocol\\_v101\\_20010615.pdf](http://www.openmobilealliance.org/tech/affiliates/syncml/syncml_protocol_v101_20010615.pdf) to Ericsson et al. (hereafter “Ericsson”) in further view of U.S. Patent Application Publication No. 2003/0220966 to Hepper et al. (hereafter “Hepper”). The Applicant respectfully disagrees.

Ong teaches a generic infrastructure for converting documents between formats with merge capabilities. The generic conversion framework allows developers to develop custom plug-in conversion algorithms and/or merge algorithms. [Ong, Abstract] Ong also teaches that the framework may provide a generic Application Programming Interface (API), through which one or more of the plug-ins may be plugged into the framework. The plug-in modules for converting, differencing and/or merging documents of various formats may interface with the framework 220 via the API. [Ong, ¶ 0139] Ong further teaches that the framework is configurable to use any of a variety of **front-ends**, for example, email, HTTP, SyncML, WebDAV, SOAP and cbXML, among others. For example, in one embodiment with an email front-end, a user of a client device may email documents to a particular synchronization implementation based on the framework, which then may email the results (synchronized documents) back to the client device. [Ong, ¶ 0143] In this example, the “front-end” application is the email application. Inclusion of SyncML and WebDAV as exemplary front-end applications indicates that the front-end application can be a synchronizing application using one of the synchronizing protocols SyncML or WebDAV. In this case, a document is sent from the client device 106 to the server 102 using the synchronization protocol. As such, Ong teaches two different synchronization processes.

The first synchronization process relates to any synchronization application that utilizes the SyncML or WebDAV protocols, which synchronizes data between two devices (the client

device 106 and the server 102 in this case). The second synchronization process relates to the synchronization and conversion process which is the subject of the Ong invention and is performed by the framework 220. The second synchronization process performs a conversion, differencing, and merging process between two differently formatted documents. Ong specifically teaches that the first synchronization process is performed by a “front-end” application, SyncML and WebDAV [Ong, ¶0143, lines 2-4], and that the second synchronization process is performed by the framework 220 and corresponding plug-in modules (Ong, ¶0143, lines 5-8). In fact, Ong teaches in ¶0143, lines 5-9 that “[f]or example, in one embodiment with an email front-end, a user of a client device may email documents to a particular synchronization implementation based on the framework 220, which then may email the results (synchronized documents) back to the client.” Two important points are made here. First, Ong makes a distinction between the “email front-end” and the “particular synchronization implementation based on the framework 220.” Substituting the front-end email application with a front-end synchronization application using a synchronization protocol (SyncML or WebDAV), there are two distinct synchronization applications. Second, Ong specifies that the results (synchronized documents) of the synchronization implementation (second synchronization process) are emailed back to the client device. This second aspect confirms the first aspect, that two separate processes are being performed, the second process where the result (synchronized documents) is generated and the first process where the result is emailed back to the server. This substantiates two separate and distinct processes, the second synchronization process and the “front-end” process, which becomes the first synchronization process when the front-end application is a synchronization application.

Within the Office Action, the generic API of Ong [Ong, ¶0139] is cited as teaching the claimed interface layer. However, the Ong generic API is an interface between the framework 220 and the plug-in modules for converting, differencing, and merging (plug-ins 210, 214, 212). As described above, “synchronization” performed by the framework 220 and the plug-in modules 210–214 is related to the second synchronization process. Within the Office Action, it is contended that Ong teaches synchronization protocols, and the SyncML and WebDAV synchronization protocols of ¶0143, lines 1-5 in Ong are cited. However, as discussed in detail above, these synchronization protocols are related to the first synchronization process, and therefore do not provide a communications means for the generic API, which provides communications within the second synchronization process. In other words, the generic API of

Ong is configured to communicate with the cited office applications and the plug-in modules as part of the second synchronization process. There is no hint, teaching, or suggestion within Ong that indicates the generic API is configured to communicate between the office applications and the synchronization protocols (front-end applications) as part of the first synchronization process. Even if the teachings of Ericsson (cited as teaching a synchronization layer) and Hepper (cited as teaching messages that are independent of the synchronization protocol) are applied to Ong, the teachings of Ericsson and Hepper are applied to the synchronization protocol of the first synchronization process, which has no bearing on the fact that the generic API of Ong is directed to the second synchronization process. Accordingly, Ong in view of Ericsson in further view of Hepper fail to teach the claimed interface layer configured to communicate with one or more applications and a synchronization layer.

In contrast with the “generic API” of Ong, the presently claimed invention teaches a generic API that is referring to a single API that is capable of communicating to multiple different synchronization protocols (a synchronization protocol stack). The generic API of the presently claimed invention is not comprised of multiple APIs that need to be called individually, instead the present invention teaches a single generic API that all communications from the application can be directed to utilizing the same format regardless of the desired synchronization protocol. Further in contrast to the teachings of Ong, the generic API of the presently claimed invention provides an interface between synchronization applications and a plurality of synchronization protocols (which can form a protocol stack). The generic synchronization API acts as an abstraction layer for the plurality of synchronization protocols. A synchronization application interfaces with the generic synchronization API in a manner independent of the specific synchronization protocol, and the generic synchronization API interfaces with each individual synchronization protocol. Unlike the presently claimed invention, Ong does not teach a generic API configured to communicate with one or more applications and a synchronization layer.

Within the Office Action, Ericsson is cited for teaching a synchronization layer which consists of synchronization protocols. Hepper is cited as teaching messages that are independent of the synchronization protocol. However, neither Ericsson nor Hepper teach a generic API configured to communicate with one or more applications and a synchronization layer and that a generic synchronization communications between the one or more applications and the interface layer are independent of a synchronization protocol used between the interface layer and the

synchronization layer. Accordingly, neither Ong, Ericsson, Hepper, nor their combination teach a generic API configured to communicate with one or more applications and a synchronization layer.

In contrast to the teachings of Ong, Ericsson, Hepper and their combination, the generic API of the presently claimed invention provides an interface between synchronization applications and a plurality of synchronization protocols. The generic synchronization API acts as an abstraction layer for the plurality of synchronization protocols. A synchronization application interfaces with the generic synchronization API in a manner independent of the specific synchronization protocol, and the generic synchronization API interfaces with each individual synchronization protocol. As described above, neither Ong, Ericsson, Hepper, nor their combination teach a generic API configured to communicate with one or more applications and a synchronization layer.

The independent Claim 1 is directed to a first device to synchronize data with a second device. The first device of Claim 1 comprises a memory comprising one or more applications, a network layer coupled to interface with the second device, a synchronization layer coupled to the network layer to provide a synchronization protocol between the first device and the second device, and an interface layer coupled to communicate with the one or more applications and the synchronization layer to provide generic synchronization communications between the one or more applications and the synchronization layer, wherein the generic synchronization communications between the one or more applications and the interface layer are independent of the synchronization protocol used between the interface layer and the synchronization layer. As discussed above, neither Ong, Ericsson, Hepper, nor their combination teach a generic API configured to communicate with one or more applications and a synchronization layer. For at least these reasons, the independent Claim 1 is allowable over the teachings of Ong, Ericsson, Hepper and their combination.

Claims 2-7 are dependent upon the independent Claim 1. As discussed above, the independent Claim 1 is allowable over the teachings of Ong, Ericsson, Hepper and their combination. Accordingly, Claims 2-7 are all also allowable as being dependent upon an allowable base claim.

The independent Claim 8 is directed to a network. The network of Claim 8 comprises one or more network devices, and an application device. The application device comprises one or

more applications, a network layer coupled to interface with the one or more network devices, a synchronization layer coupled to the network layer to provide a synchronization protocol between the application device and the one or more network devices, and an interface layer coupled to communicate with the one or more applications and the synchronization layer to provide generic synchronization communications between the one or more applications and the synchronization layer, wherein the generic synchronization communications between the one or more applications and the interface layer are independent of any synchronization protocol used between the interface layer and the synchronization layer. As discussed above, neither Ong, Ericsson, Hepper, nor their combination teach a generic API configured to communicate with one or more applications and a synchronization layer. For at least these reasons, the independent Claim 8 is allowable over the teachings of Ong, Ericsson, Hepper and their combination.

Claims 9-13 are dependent upon the independent Claim 8. As discussed above, the independent Claim 8 is allowable over the teachings of Ong, Ericsson, Hepper and their combination. Accordingly, Claims 9-13 are all also allowable as being dependent upon an allowable base claim.

The independent Claim 15 is directed to a method of providing an interface to one or more synchronization applications resident within a first device coupled to a network of devices. The method of Claim 15 comprises sending and receiving messages to and from the one or more synchronization applications through an interface layer to one or more synchronization protocol stacks, to synchronize data between the first device and at least one other device within the network of devices, wherein the messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization protocol stacks, and generating and receiving communications at the interface layer to complete data synchronization between the first device and the at least one other device within the network of devices. As discussed above, neither Ong, Ericsson, Hepper, nor their combination teach a generic API configured to communicate with one or more applications and a synchronization layer. For at least these reasons, the independent Claim 15 is allowable over the teachings of Ong, Ericsson, Hepper and their combination.

Claims 16-18 and 20 are dependent upon the independent Claim 15. As discussed above, the independent Claim 15 is allowable over the teachings of Ong, Ericsson, Hepper and their combination. Accordingly, Claims 16-18 and 20 are all also allowable as being dependent upon an allowable base claim.

The independent Claim 21 is directed to an apparatus for providing an interface to one or more synchronization applications resident within a first device coupled to a network of devices. The apparatus of Claim 21 comprises a memory comprising means for sending and receiving messages to and from the one or more synchronization applications through an interface layer to one or more synchronization protocol stacks, to synchronize data between the first device and at least one other device within the network of devices, wherein the messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization protocol stacks, and means for generating and receiving communications at the interface layer to complete data synchronization between the first device and the at least one other device within the network of devices. As discussed above, neither Ong, Ericsson, Hepper, nor their combination teach a generic API configured to communicate with one or more applications and a synchronization layer. For at least these reasons, the independent Claim 21 is allowable over the teachings of Ong, Ericsson, Hepper and their combination.

Claims 22-24 are dependent upon the independent Claim 21. As discussed above, the independent Claim 21 is allowable over the teachings of Ong, Ericsson, Hepper and their combination. Accordingly, Claims 22-24 are all also allowable as being dependent upon an allowable base claim.

Within the Office Action, Claims 7, 14, 19 and 25 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Ong in view of Ericsson in view of Hepper and in further view of U.S. Patent Application No. 2003/0014483 to Stevenson et al. (hereafter "Stevenson"). Applicants respectfully disagree.

Claims 7, 14, 19 and 25 are dependent on the independent Claims 1, 8, 15 and 21, respectively. As described above, the independent Claims 1, 8, 15 and 21 are all allowable over Ong, Ericsson, Hepper and their combination. Accordingly, Claims 7, 14, 19 and 25 are all also allowable as being dependent upon an allowable base claim.

For at least the reasons given above, the Applicant respectfully submits that the claims are in a condition for allowance, and allowance at an early date would be appreciated. Should the Examiner have any questions or comments, the Examiner is encouraged to call the undersigned at (408) 530-9700 to discuss the same so that any outstanding issues can be expeditiously resolved.

Respectfully submitted,  
HAVERSTOCK & OWENS LLP

Dated: February 25, 2009

By: /Jonathan O. Owens/

Jonathan O. Owens  
Reg. No. 37,902  
Attorneys for Applicant